

Manual de Utilização da Plataforma SIGUS

Coordenador: Hemerson Pistori

Manual desenvolvido no âmbito do projeto "Plataforma de Apoio ao Desenvolvimento de Sistemas para Inclusão Digital de Pessoas com Necessidades Especiais" apoiado pela FINEP, Convênio 01.05.0710.00 – Ref. 2161/05

Campo Grande, MS

Sumário

1. Objetivos do Manual

Este manual apresenta orientações para técnicos e profissionais com formação em computação de como utilizar a plataforma SIGUS para o desenvolvimento de interfaces guiadas por sinais visuais. Não faz parte do escopo deste manual a instalação de softwares básicos como sistemas operacionais e ambiente de programação (como um ambiente Java, por exemplo, requisito para da plataforma SIGUS). Manuais para isto são facilmente encontrados na Internet.

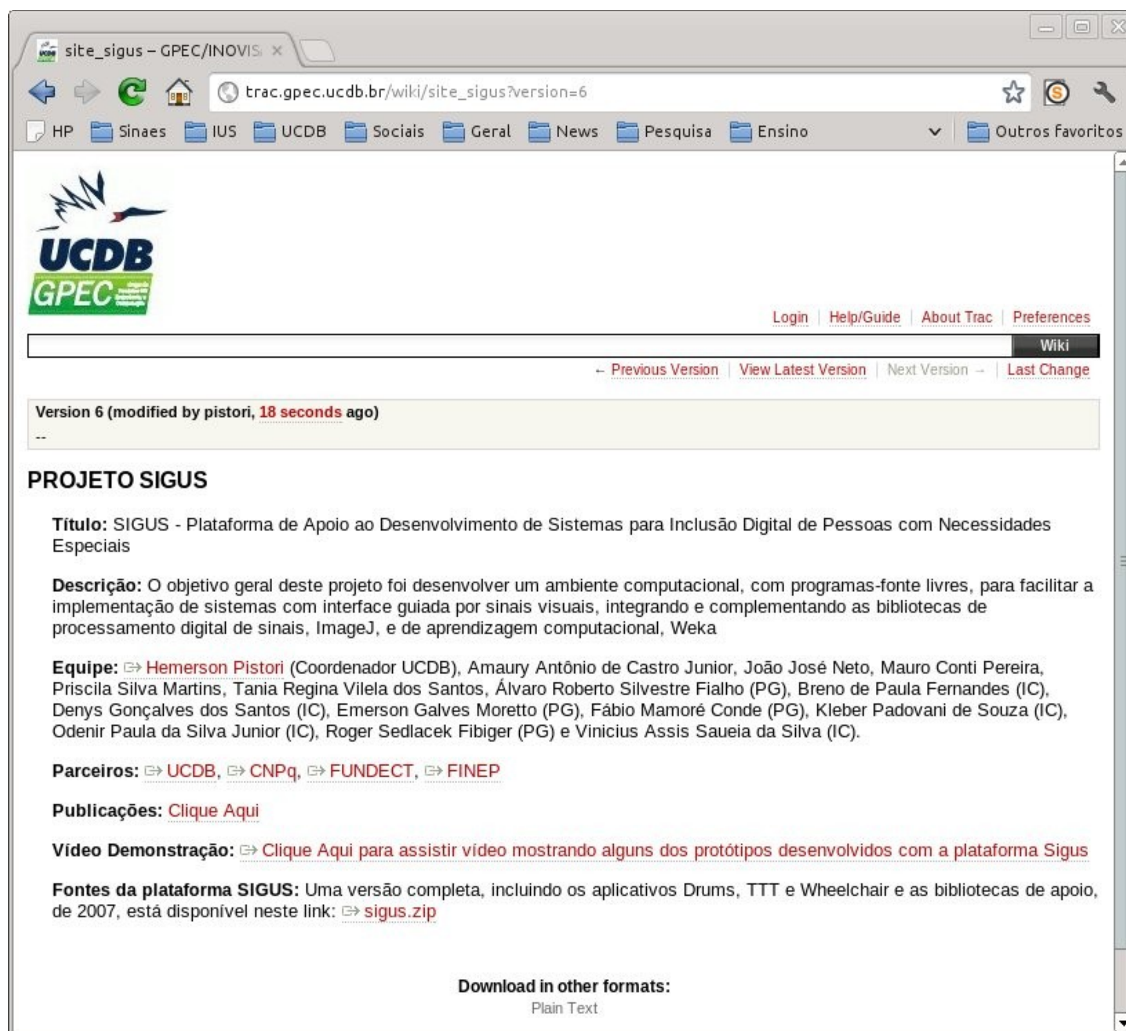
2. Requisitos para utilização

Este manual foi desenvolvido entre os anos de 2005 e 2007 e portanto, as bibliotecas auxiliares e sistemas operacionais testados correspondem às versões disponíveis na época. As soluções podem não funcionar em versões posteriores e diferentes daquelas utilizadas, que foram:

- Sistemas Operacionais: Linux Debian 3.1 (Kernel 2.6.8-2-386)
- Java 2 Platform, SE, v1.4.2
- JMF 2.1.1.e
- Plataforma SIGUS disponível em http://www.gpec.ucdb.br/pistori/projetos/sigus/sigus.zip
- Webcam compatível com JMF 2.1.1. Exemplares testados:
    - Quickcam for Notebooks Pro
    - Quicam Pro 4000
    - EasyCam Pro HO98064
    - Genius Video Cam Série V4

Observação importante: o sistema foi criado em Java visando a máxima portabilidade entre sistemas operacionais, tendo sido testado em 2007 também nas plataformas Windows XP e

MacOS (hardware e versões disponíveis na época). A tela abaixo mostra a página do site onde a plataforma SIGUS pode ser obtida gratuitamente (último link em vermelho do site com a palavra "sigus.zip"). O arquivo deve ser descompactado utilizando programas compatíveis com o formato .zip (E.g: winzip, linux zip, etc.).



Recomendamos fortemente a leitura do *Manual de Utilização de Webcam no Desenvolvimento de Aplicativos Java*, também disponível no site do projeto SIGUS, antes da utilização da plataforma, pois esta depende da existência de uma webcam configurada corretamente com o JMF para funcionar.

3. Instalação da Plataforma SIGUS

3.1 Download da Plataforma:

A plataforma pode ser obtida através do site do projeto SIGUS, mostrado na seção anterior, ou diretamente através deste link: http://www.gpec.ucdb.br/pistori/projetos/sigus/sigus.zip. O arquivo deve ser descompactado utilizando algum software compatível com o formato .zip (E.g: winzip ou zip para Linux).

3.2 Estrutura de pastas e principais arquivos

Após descompactar o arquivo sigus.zip será gerada uma estrutura de pastas em seu computador contendo centenas de arquivos, muitos com exemplos e utilitários complementares ao projeto SIGUS, além de todos os fontes e documentação do sistema. Neste manual de utilização, iremos nos concentrar nas pastas e arquivos essencial para a utilização da plataforma. A figura abaixo mostra a estrutura principal de pastas da plataforma SIGUS que será explicada a seguir.

```
svn : bash
Arquivo  Editar  Exibir  Favoritos  Configurações  Ajuda
svn : bash

~/Downloads/svn$ ls
applications  libraries  README.htm  utils
~/Downloads/svn$
~/Downloads/svn$ ls applications
drums  genius  libras  README.htm  ttt  wheelchair
~/Downloads/svn$
~/Downloads/svn$ ls libraries
adaptmap         imagej                  jama          jfreechart  ncsa        README.htm    spring
adaptools        internationalisation    jannealer1.1  jmathtools  openjgraph  sigus         swinglabs
colorInspector   itext                   java3d        JMF-2.1.1e  pega        sigus-maven   swing-layout
comirva          jahmm-0.6.1             jdbc          log4j       poi-3.0.2   skinlf        weka
hibernate        jakarta-commons         jdom          mical       random      soundplayer
~/Downloads/svn$
~/Downloads/svn$ 
```

No exemplo mostrado na figura, o arquivo sigus.zip foi descompactado dentro de uma pasta

chamada “Downloads”. Dentro da pasta “Downloads” é criada uma pasta chamada “svn” indicando que a plataforma é mantida através do sistema de controle de versões o subversion http://subversion.tigris.org/.

Cada pasta, que são mostradas em azul, contém um arquivo “README.htm” que pode ser aberto através de um navegador de Internet e que resume o conteúdo da pasta. Na raiz da estrutura de pastas do SIGUS estão as pastas “applications”, “libraries” e “utils”. Destas, as duas primeiras são as mais importantes para a utilização da plataforma, e a terceira interessa mais aos programadores que queiram aprimorar a própria plataforma e testar novos algoritmos.



3.2.1 Pasta Applications

É na pasta “applications” que ficam os protótipos e exemplos de softwares desenvolvidos através da plataforma SIGUS. Por exemplo, o aplicativo “drums” é uma bateria musical virtual que pode ser acionada com movimentos da face. Já o aplicativo “ttt” é o Tic-Tac-Toe (jogo da velha). A pasta “libras” contém o protótipo de editor que reconhece o alfabeto LIBRAS sinalizado pelas mãos e a “wheelchair” o protótipo de um sistema para controle de direção de cadeiras de rodas utilizando movimentos da cabeça. Um aplicativo que emita o jogo “Genius” para ser jogado com movimentos do braço está disponível na pasta “genius”. Um vídeo no youtube mostrando alguns

destes aplicativos em ação está disponível neste endereço: http://youtu.be/wwrHOgLHKFQ

Cada pasta de aplicativo segue uma mesma estrutura que será apresentada na Seção 4 deste manual.

3.2.2 Pasta Libraries

É nesta pasta que ficam os fontes da plataforma SIGUS, além de diversas bibliotecas de apoio, como imagej, weka e JMF que foram integradas através da plataforma. A mais importante sub-pasta da pasta "libraries" é justamente a pasta "sigus". Em especial, o arquivo sigus.jar, que é a plataforma pronta para ser utilizada na construção de aplicativos e o arquivo "index.html", que contém a documentação javadoc das classes que compõe a biblioteca, são de interesse particular para os desenvolvedores. Como trata-se de uma documentação em html, recomenda-se a utilização de um navegador de internet para abrir o arquivo index.html. A utilização do arquivo sigus.jar é explicada na Seção 4 deste manual. O caminho completo para estes dois arquivos, assumindo a descompactação na pasta Downloads é:

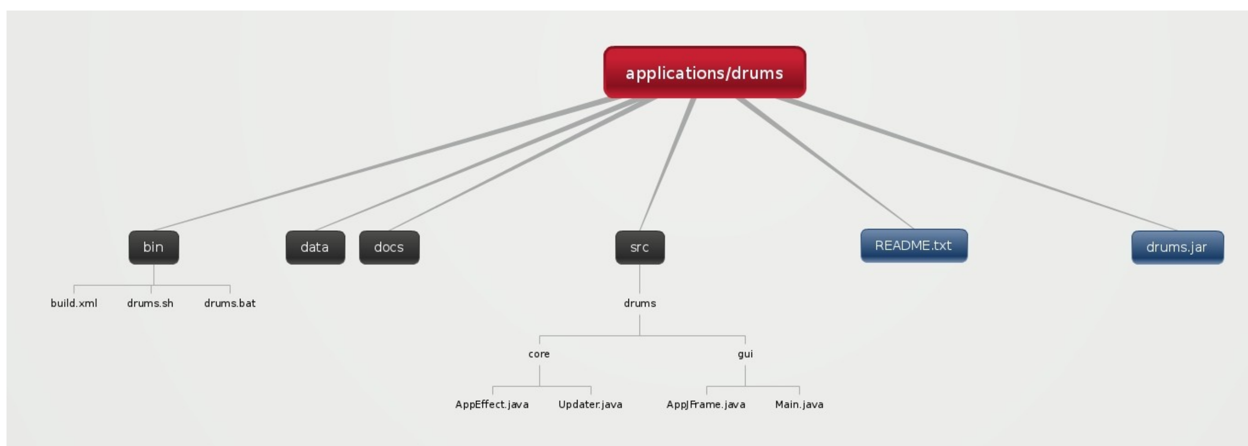
$ Downloads/svn/libraries/sigus/sigus.jar

$ Downloads/svn/libraries/sigus/docs/javadoc/index.html

4. Criação de Aplicativos

O primeiro passo para se criar uma nova aplicação utilizando SIGUS e ter acesso à sua interface gráfica (mostrada na próxima seção deste manual) é escolher um dos aplicativos já existentes como modelo, replicá-lo em uma nova pasta e ajustar alguns parâmetros de compilação. Nesta seção iremos detalhar este procedimento.

4.1. Estrutura de pastas de um aplicativo SIGUS



A figura acima resume a estrutura de pastas e arquivos de um aplicativo baseado na plataforma SIGUS. Neste exemplo, estamos utilizando o aplicativo drums que fica dentro da pasta svn/applications. A pasta “bin” é onde ficam os arquivos executáveis e os arquivos auxiliares para geração dos executáveis utilizando o ant (build.xml) e para a inicialização do aplicativo (drums.sh para ambientes Linux e drums.bat para ambientes Windows). Na pasta “data” estão arquivos de imagens e sons utilizados pelo aplicativo, quando for o caso. No exemplo drums, esta pasta contém as imagens das peças da bateria e seus respectivos sons (E.g: bumbo.jpg, caixa.jpg, chimbalAberto.wav, prato.wav, etc). A pasta “docs” é reservada para arquivos de documentação

e a pasta “src”, a principal delas, é onde ficam os fontes do aplicativo. São 4 os arquivos-fonte de um aplicativo, que se dividem em dois grupos:

(1) Core: programas responsáveis pela integração entre os módulos de captura de imagens (JMF), processamento de imagens (IMAGEJ) e aprendizagem automática (WEKA). Estes programas raramente precisam modificados de aplicativo para aplicativo.

(2) Gui: programas que implementam a interface e a lógica específica do aplicativo. O fonte “AppJFrame.java” é onde os detalhes de cada aplicativo ficam e portanto, é onde as modificações para o novo aplicativo devem estar.

4.2. Replicação e ajustes para novos aplicativos

Para replicar um dos aplicativos, por exemplo, “drums” e criar um novo chamado, por exemplo “guitar”, a seguinte sequência de comandos pode ser executada (comentários entre colchetes):

$ cd ~/Downloads/svn/applications/   [entrar na pasta de aplicativos]

$ mkdir guitar   [criar pasta para o novo aplicativo]

$ cp -r drums/* guitar   [copiar arquivos do aplicativo drums para o novo aplicativo]

$ cd guitar   [entrar na pasta do novo aplicativo]

$ ls   [listar arquivos da pasta e verificar se foram realmente copiados]

Após a replicação, é preciso alterar, dentro da pasta “guitar” as ocorrências da palavra “drums” por “guitar”. Para não entrar em cada arquivo, recomendamos a utilização de uma ferramenta

como o Eclipse. Um tutorial de como fazer isto no Eclipse esta disponível aqui: http://www.avajava.com/tutorials/lessons/how-do-i-do-a-find-and-replace-in-multiple-files-in-eclipse.html

Caso o link esteja quebrado quando da leitura deste manual, basta buscar por “find and replace multiple files eclipse” em qualquer ferramenta de busca na Internet que várias opções serão retornadas. Altere também os nomes das pastas que forem “drums”, como src/drums, para “guitar”. Para finalizar, caso o seu aplicativo vá utilizar uma quantidade de gestos ou sinais diferente do aplicativo modelo, é preciso realizar os seguintes ajustes:

(1) Dentro do método init() da classe Updater(), em src/guitar/update.java, troque o valor inicial da variável “classValue” pelo total de gestos que será utilizado no novo aplicativo. Altere também, de forma correspondente, a inicialização da variável “arffClasses”.

(2) Dentro do método init() da classes AppEffect, em src/guitar/AppEffect.java, troque o valor inicial da variável MAX_CLASS pelo total de gestos que será utilizado no novo aplicativo somado de 1 (ou seja, se forem 4 gestos, coloque 5 na variável MAX_CLASS).

A interface e a lógica do novo aplicativo é escrita livremente em Java e deve ser implementada através do fonte src/guitar/AppJFrame (ou de classes novas, mas que precisam ser referenciadas e inicialização a partir da AppJFrame.

4.3. Compilação e execução

A compilação e execução dos aplicativos podem ser feitas utilizando qualquer ambiente que suporte a programação em Java. Neste tutorial, iremos tratar da estratégia que foi utilizada pelos desenvolvedores e que tem como vantagens a portabilidade. Tanto para compilação da própria plataforma, quando dos aplicativos, foi utilizado o Apache Ant (http://ant.apache.org/).

4.3.1 Compilação do aplicativo *guitar*

A seguinte sequência de comandos pode ser utilizada para compilar o novo aplicativo:

$ cd ~/Downloads/svn/applications/guitar/bin   [entrar na pasta de binários do novo aplicativo]

$ ant   [gera o arquivo guitar.jar agregando as classes compiladas e os metadados]

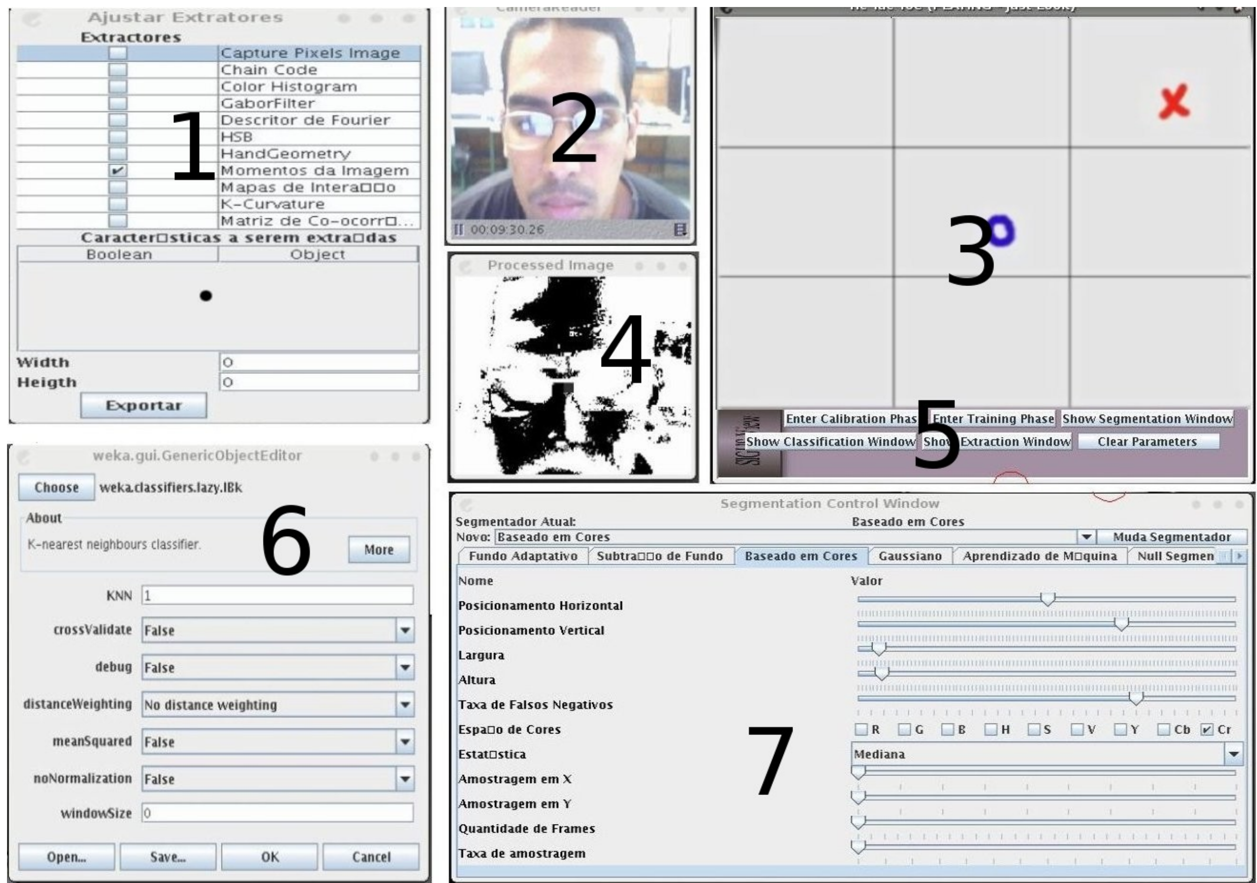
Caso o guitar.jar não seja gerado, confira o arquivo build.xml, utilizado pelo ant, para se certificar de que as variáveis estão corretas. Note que o build.xml utiliza caminhos relativos (relative paths) para facilitar a reutilização, no entanto, é preciso respeitar a hierarquia de pastas da distribuição da plataforma SIGUS, ou então realizar os devidos ajustes dentro de build.xml. Também podem ser necessários ajustes para diferentes plataformas e versões mais novas da ferramenta ant, do java, do sistema operacional e das bibliotecas auxiliares.

4.3.2 Execução do aplicativo *guitar*

Caso o ant tenha sido executado sem problemas, basta utilizar agora o script “guitar.sh” para iniciar o aplicativo e a plataforma sigus: $ sh ./guitar.sh

Em caso de problemas, abra este arquivo, que está na pasta bin também, e realize os devidos ajustes. Note que é feita uma referencia a um arquivo, responsável pelo ajuste da variável classpath do Java, que fica na pasta ~/Downloads/svn/utils/linux/scripts/setClassPath.sh. Programadores Java não deverão encontrar dificuldades para atualizar o classpath tendo como base o build.xml ou então o script setClassPath.sh que acompanha a plataforma.

5. Interface Gráfica da Plataforma SIGUS



A Figura acima mostra os 7 elementos principais da Interface Gráfica da plataforma SIGUS. Cada elemento, indicado com um número grande (1 a 7), será descrito a seguir:

5.1. Janela para ajuste dos extratores

Nesta janela o desenvolver determina quais os algoritmos que serão utilizados na extração de atributos. Diferentes algoritmos podem ser combinados e dependendo do algoritmo, pode ser necessário ainda ajustar algumas características ou parâmetros. O botão “exportar” deve ser utilizado depois que os algoritmos forem escolhidos. Como mostrado na janela, entre os algoritmos disponíveis estão o Chain Code, Gabor Filter, Momentos da Imagem e o K-Curvatura.

5.2. Janela de captura de imagens

Esta é a janela que mostra em tempo real as imagens que estão sendo capturadas pela webcam. Ela também pode apresentar a janela de seleção utilizada por alguns dos algoritmos de segmentação de pele humana utilizados no pré-processamento das imagens. A janela de seleção é um pequena região da imagem, escolhida pelo usuário (através do movimento de arrastar com o mouse), contendo uma amostra de pele humano. Alguns dos algoritmos utilizados no SIGUS não precisam de calibração e caberá ao desenvolver determinar qual o mais adequado dependendo do problema. Geralmente, a calibração precisa ser realizada uma única vez, pelo usuário ou por algum responsável por ajudar o usuário (em caso de deficiência motora nas mãos).

5.3. Janela de Interface do Aplicativo

A janela de interface é específica para cada novo aplicativo. Neste exemplo, temos um jogo da velha que pode ser jogado com movimentos da face associados a cada posição do tabuleiro.

5.4. Janela de resultado de processamento

Esta janela mostra o resultado dos algoritmos de segmentação que é uma imagem binarizada (preto e branco) mostrando em preto as regiões que irão ser transmitidas para os próximos módulos: extração de atributos e aprendizagem automática.

5.5. Janela principal da plataforma SIGUS

Através desta janela o desenvolvedor ou usuário controla o módulo que está ativo a cada momento e as janelas que devem ser apresentadas. É importante lembrar que a plataforma SIGUS implementa um tipo de interface bastante flexível baseada em aprendizagem automática que depende de uma fase de treinamento onde os gestos e sinais realizados pelo usuário (faces, mãos, dedos, olhos, pernas, etc) devem ser associados a ações na interface. No exemplo mostrado na Figura desta seção, 9 diferentes

sinais devem ser realizados e associados a cada uma das nove posições existentes no tabuleiro do jogo da velha. Dependendo da aplicação, uma quantidade diferente de sinais deve ser utilizada. Para cada aplicação e usuário, uma nova fase de treinamento deve ser realizada. Os seguintes botões estão disponíveis nesta janela:

Enter Calibration Phase: este botão deve ser clicado para se voltar ou entrar na fase de calibração do segmentador, que consiste em selecionar uma região da imagem capturada pela webcam que contenha pele humana (nem todos os segmentadores disponíveis necessitam de calibração). O mesmo botão, que mudará de nome para "exit calibration phase" deve ser usado para sair da fase de calibração.

Enter Training Phase: este botão deve ser clicado para se voltar ou entrar na fase de treinamento, quando os gestos devem ser efetuados (E.g: olhar para uma determinada região do tabuleiro) ao mesmo tempo em que a ação desejada (E.g: clicar em uma determinada região do tabuleiro do jogo da velha) é realizada. É importante que isto seja feito várias vezes para que o sistema "aprenda" as diferentes nuances dos gestos que serão efetuados. Como são utilizadas técnicas estatísticas de aprendizagem supervisionada, de forma geral, quanto mais exemplos melhor o sistema irá funcionar. O mesmo botão, que mudará de nome para "exit training phase" deve ser usado para sair da fase de treinamento. Ao sair da fase de treinamento, o sistema levará alguns segundos para realizar a aprendizagem dos novos exemplos, e o usuário passará a poder controlar a interface do aplicativo usando gestos ou movimentos (ou seja, sem precisar do mouse ou do teclado)

Show segmentation window: Mostra a janela que permite escolher e ajustar os parâmetros dos algoritmos de segmentação

Show classification window: Mostra a janela que permite escolher e ajustar os parâmetros dos algoritmos de aprendizagem automática

Show extraction window: Mostra a janela que permite escolher e ajustar os parâmetros dos algoritmos de extração de atributos (ou extração de características)

Clear parameters: Limpa o conjunto de treinamento capturado na fase de treinamento para que um novo conjunto possa ser gerado. Este botão é geralmente utilizado quando um novo usuário começa a utilizar o sistema e os resultados indicam que um novo treinamento deve ser realizado para que o sistema se readapte

5.6. Janela para ajuste do módulo de aprendizagem automática

É nesta janela que se determina o algoritmo de aprendizagem automática que será utilizado (botão "choose"). Como o SIGUS está integrado ao Weka, todos os algoritmos do Weka podem ser utilizados. Ao escolher o algoritmo, um conjunto de parâmetros, específicos do algoritmo, podem ser ajustados. Depois dos ajustes, o botão "OK" deve ser clicado para que a troca de algoritmo seja efetuada.

5.7. Janela para ajuste do módulo de segmentação

Nesta janela é possível escolher qual a técnica de segmentação a ser adotada e também ajustar os parâmetros referentes a cada técnica.